Week 11: Capstone Project Part 4 NUS Salman

```
!pip install -U langchain langchain-community
!pip install pymupdf
```

```
Requirement already satisfied: langchain in /usr/local/lib/python3.11/dist-packages (0.3.26)
Requirement already satisfied: langchain-community in /usr/local/lib/python3.11/dist-packages (0.3.27)
Requirement already satisfied: langchain-core<1.0.0,>=0.3.66 in /usr/local/lib/python3.11/dist-packages (from langchain) (0.3.69)
Requirement already satisfied: langchain-text-splitters<1.0.0,>=0.3.8 in /usr/local/lib/python3.11/dist-packages (from langchain) (0.3.8)
Requirement already satisfied: langsmith>=0.1.17 in /usr/local/lib/python3.11/dist-packages (from langchain) (0.4.6)
Requirement already satisfied: pydantic<3.0.0,>=2.7.4 in /usr/local/lib/python3.11/dist-packages (from langchain) (2.11.7)
Requirement already satisfied: SQLAlchemy<3,>=1.4 in /usr/local/lib/python3.11/dist-packages (from langchain) (2.0.41)
Requirement already satisfied: requests<3,>=2 in /usr/local/lib/python3.11/dist-packages (from langchain) (2.32.3)
Requirement already satisfied: PyYAML>=5.3 in /usr/local/lib/python3.11/dist-packages (from langchain) (6.0.2)
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in /usr/local/lib/python3.11/dist-packages (from langchain-community) (3.11.15)
Requirement already satisfied: tenacity!=8.4.0,<10,>=8.1.0 in /usr/local/lib/python3.11/dist-packages (from langchain-community) (8.5.0)
Requirement already satisfied: dataclasses-json<0.7,>=0.5.7 in /usr/local/lib/python3.11/dist-packages (from langchain-community) (0.6.7)
Requirement already satisfied: pydantic-settings<3.0.0,>=2.4.0 in /usr/local/lib/python3.11/dist-packages (from langchain-community) (2.10.1)
Requirement already satisfied: httpx-sse<1.0.0,>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from langchain-community) (0.4.1)
Requirement already satisfied: numpy>=1.26.2 in /usr/local/lib/python3.11/dist-packages (from langchain-community) (2.0.2)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain-community) (2.6.1)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain-community) (1.4.0)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain-community) (25.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain-community) (1.7.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain-community) (6.6.3)
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain-community) (0.3.2)
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain-community) (1.20.1)
Requirement already satisfied: marshmallow<4.0.0,>=3.18.0 in /usr/local/lib/python3.11/dist-packages (from dataclasses-json<0.7,>=0.5.7->langchain-community) (3.26.1)
Requirement already satisfied: typing-inspect<1,>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from dataclasses-json<0.7,>=0.5.7->langchain-community) (0.9.0)
Requirement already satisfied: jsonpatch<2.0,>=1.33 in /usr/local/lib/python3.11/dist-packages (from langchain-core<1.0.0,>=0.3.66->langchain) (1.33)
Requirement already satisfied: typing-extensions>=4.7 in /usr/local/lib/python3.11/dist-packages (from langchain-core<1.0.0,>=0.3.66->langchain) (4.14.1)
Requirement already satisfied: packaging>=23.2 in /usr/local/lib/python3.11/dist-packages (from langchain-core<1.0.0,>=0.3.66->langchain) (25.0)
Requirement already satisfied: httpx<1,>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from langsmith>=0.1.17->langchain) (0.28.1)
Requirement already satisfied: orjson<4.0.0,>=3.9.14 in /usr/local/lib/python3.11/dist-packages (from langsmith>=0.1.17->langchain) (3.11.0)
Requirement already satisfied: requests-toolbelt<2.0.0,>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from langsmith>=0.1.17->langchain) (1.0.0)
Requirement already satisfied: zstandard<0.24.0,>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from langsmith>=0.1.17->langchain) (0.23.0)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<3.0.0,>=2.7.4->langchain) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.11/dist-packages (from pydantic<3.0.0,>=2.7.4->langchain) (2.33.2)
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<3.0.0,>=2.7.4->langchain) (0.4.1)
Requirement already satisfied: python-dotenv>=0.21.0 in /usr/local/lib/python3.11/dist-packages (from pydantic-settings<3.0.0,>=2.4.0->langchain-community) (1.1.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2->langchain) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2->langchain) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2->langchain) (2.4.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2->langchain) (2025.7.14)
Requirement already satisfied: greenlet>=1 in /usr/local/lib/python3.11/dist-packages (from SQLAlchemy<3,>=1.4->langchain) (3.2.3)
Requirement already satisfied: anyio in /usr/local/lib/python3.11/dist-packages (from httpx<1,>=0.23.0->langsmith>=0.1.17->langchain) (4.9.0)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packages (from httpx<1,>=0.23.0->langsmith>=0.1.17->langchain) (1.0.9)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/dist-packages (from httpcore==1.*->httpx<1,>=0.23.0->langsmith>=0.1.17->langchain) (0.16.0)
Requirement already satisfied: jsonpointer>=1.9 in /usr/local/lib/python3.11/dist-packages (from jsonpatch<2.0,>=1.33->langchain-core<1.0.0,>=0.3.66->langchain) (3.0.0)
Requirement already satisfied: mypy-extensions>=0.3.0 in /usr/local/lib/python3.11/dist-packages (from typing-inspect<1,>=0.4.0->dataclasses-json<0.7,>=0.5.7->langchain-community) (1.1.0)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-packages (from anyio->httpx<1,>=0.23.0->langsmith>=0.1.17->langchain) (1.3.1)
Collecting pymupdf
  Downloading pymupdf-1.26.3-cp39-abi3-manylinux_2_28_x86_64.whl.metadata (3.4 kB)
Downloading pymupdf-1.26.3-cp39-abi3-manylinux_2_28_x86_64.whl (24.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.1/24.1 MB 72.1 MB/s eta 0:00:00
Installing collected packages: pymupdf
Successfully installed pymupdf-1.26.3
```

```
from langchain_community.document_loaders import PyMuPDFLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter
```

Task 1: Document Loading and Preprocessing

```
from langchain_community.document_loaders import PyMuPDFLoader

# Load the sample PDF file
pdf_path = "HR_Policy_Sample.pdf"
loader = PyMuPDFLoader(pdf_path)
documents = loader.load()

# Display the first chunk (if available)
print("First document chunk:\n")
print(documents[0].page_content[:500])


# Load the PDF
pdf_path = "HR_Policy_Sample.pdf"  # Upload this file in Colab if not already
loader = PyMuPDFLoader(pdf_path)
documents = loader.load()

#  Split into chunks of ~500 tokens
text_splitter = RecursiveCharacterTextSplitter(
    chunk_size=300,
    chunk_overlap=50
)
chunks = text_splitter.split_documents(documents)

# Display first 3 representative chunks
print("✅ Total Chunks:", len(chunks), "\n")
for i in range(3):
    print(f"📄 Chunk {i+1}:\n")
    print(chunks[i].page_content[:500])  # Show up to 500 characters
    print("-" * 80)
```

```
First document chunk:

Company HR Policy Sample
1. Attendance and Punctuality
Employees are expected to report to work on time and maintain regular attendance. Unscheduled
absences must be communicated to the manager.
2. Code of Conduct
All employees must behave professionally and respectfully in the workplace. Harassment or
discrimination of any kind will not be tolerated.
3. Leave Policy
Employees are entitled to paid leave as per the number of years served in the organization. All
leave must be approved by the repo
✅ Total Chunks: 4

📄 Chunk 1:

Company HR Policy Sample
1. Attendance and Punctuality
Employees are expected to report to work on time and maintain regular attendance. Unscheduled
absences must be communicated to the manager.
```

```
2. Code of Conduct
--------------------------------------------------------------------------------
📄 Chunk 2:

2. Code of Conduct
All employees must behave professionally and respectfully in the workplace. Harassment or
discrimination of any kind will not be tolerated.
3. Leave Policy
Employees are entitled to paid leave as per the number of years served in the organization. All
--------------------------------------------------------------------------------
📄 Chunk 3:

leave must be approved by the reporting manager in advance.
4. Remote Work
Remote work is permitted for eligible employees subject to manager approval and business needs.
5. Performance Reviews
Performance evaluations will be conducted annually. Constructive feedback will be provided and
--------------------------------------------------------------------------------
```

Task 2: Text Embedding and Vector Store Setup

```
!pip install -q faiss-cpu sentence-transformers


from langchain.embeddings import HuggingFaceEmbeddings
from langchain.vectorstores import FAISS
from langchain.vectorstores.faiss import FAISS
import os


embedding_model = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")


vector_store = FAISS.from_documents(chunks, embedding_model)


query = "What is the company’s leave policy?"
results = vector_store.similarity_search(query, k=3)


print(f"🔍 Query: {query}\n")
for i, res in enumerate(results, 1):
    print(f"📄 Result {i}:\n")
    print(res.page_content[:500])
    print("-" * 80)
```

```
                                        ━━━━━━━━━━━━━━━━━━━━ 31.3/31.3 MB 39.2 MB/s eta 0:00:00
                                        ━━━━━━━━━━━━━━━━━━━━ 363.4/363.4 MB 1.4 MB/s eta 0:00:00
                                        ━━━━━━━━━━━━━━━━━━━━ 13.8/13.8 MB 43.8 MB/s eta 0:00:00
                                        ━━━━━━━━━━━━━━━━━━━━ 24.6/24.6 MB 10.1 MB/s eta 0:00:00
                                        ━━━━━━━━━━━━━━━━━━━━ 883.7/883.7 kB 24.4 MB/s eta 0:00:00
                                        ━━━━━━━━━━━━━━━━━━━━ 664.8/664.8 MB 721.1 kB/s eta 0:00:00
                                        ━━━━━━━━━━━━━━━━━━━━ 211.5/211.5 MB 5.2 MB/s eta 0:00:00
                                        ━━━━━━━━━━━━━━━━━━━━ 56.3/56.3 MB 12.1 MB/s eta 0:00:00
                                        ━━━━━━━━━━━━━━━━━━━━ 127.9/127.9 MB 7.3 MB/s eta 0:00:00
                                        ━━━━━━━━━━━━━━━━━━━━ 207.5/207.5 MB 8.1 MB/s eta 0:00:00
                                        ━━━━━━━━━━━━━━━━━━━━ 21.1/21.1 MB 88.2 MB/s eta 0:00:00
54748.py:10: LangChainDeprecationWarning: The class `HuggingFaceEmbeddings` was deprecated in LangChain 0.2.2 and will be removed in 1.0. An updated version of the class exists in the :class:`~langchain-huggingface package and should be used instead. To use it run `pip install -U :class:`~langchain-huggingface` and import as `from :class:`~langchain_huggingface impor
gFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")
dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
 not exist in your Colab secrets.
ugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.
this secret in all of your notebooks.
ation is recommended but still optional to access public models or datasets.

                                        349/349 [00:00<00:00, 28.7kB/s]

ɔ: 100%                                 116/116 [00:00<00:00, 7.78kB/s]

)0<00:00, 579kB/s]

                                        53.0/53.0 [00:00<00:00, 3.84kB/s]

                                    612/612 [00:00<00:00, 40.1kB/s]

                                        90.9M/90.9M [00:01<00:00, 77.8MB/s]

                                        350/350 [00:00<00:00, 19.1kB/s]

):00, 7.02MB/s]

)0<00:00, 16.2MB/s]

                                        112/112 [00:00<00:00, 9.87kB/s]

                                    190/190 [00:00<00:00, 13.7kB/s]
ɔany’s leave policy?



professionally and respectfully in the workplace. Harassment or
 will not be tolerated.

paid leave as per the number of years served in the organization. All
--------------------------------------------------------------------------------


ity
report to work on time and maintain regular attendance. Unscheduled
ted to the manager.

--------------------------------------------------------------------------------


the reporting manager in advance.

or eligible employees subject to manager approval and business needs.

ll be conducted annually. Constructive feedback will be provided and
--------------------------------------------------------------------------------
```

Task 3: Retrieval-Augmented Generation with Conversational Memory

```
!pip install -q langchain langchain-google-genai chromadb pymupdf


import os
from langchain_community.document_loaders import PyMuPDFLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain_google_genai import GoogleGenerativeAIEmbeddings, ChatGoogleGenerativeAI
from langchain.vectorstores import Chroma
from langchain.chains import RetrievalQA


GOOGLE_API_KEY = "AIzaSyAiktBgA2IKDN1u7x_JJzStnK0TELsc2KE"
os.environ["GOOGLE_API_KEY"] = GOOGLE_API_KEY


pdf_path = "HR_Policy_Sample.pdf"
loader = PyMuPDFLoader(pdf_path)
documents = loader.load()

splitter = RecursiveCharacterTextSplitter(chunk_size=500, chunk_overlap=50)
chunks = splitter.split_documents(documents)


embedding_model = GoogleGenerativeAIEmbeddings(
    model="models/embedding-001",
    google_api_key=GOOGLE_API_KEY
)

vectordb = Chroma.from_documents(
    documents=chunks,
    embedding=embedding_model,
    collection_name="hr_policy_docs"
)


llm = ChatGoogleGenerativeAI(
    model="gemini-1.5-flash-latest",
    temperature=0.5
)

retriever = vectordb.as_retriever()

qa_chain = RetrievalQA.from_chain_type(
    llm=llm,
    retriever=retriever,
    return_source_documents=True,
    input_key="query"
)
```

```
queries = [
    "What is the leave policy?",
    "Is there any probation period mentioned?",
    "How many leaves are allowed during probation?"
]

for q in queries:
    result = qa_chain.invoke({"query": q})
    print(f"👤 User: {q}\n🤖 Assistant: {result['result']}\n{'-'*60}")
```

```
👤 User: What is the leave policy?
🤖 Assistant: Employees are entitled to paid leave based on their years of service.  All leave must be approved in advance by the reporting manager.
------------------------------------------------------------
👤 User: Is there any probation period mentioned?
🤖 Assistant: No, there is no mention of a probationary period in the provided text.
------------------------------------------------------------
👤 User: How many leaves are allowed during probation?
🤖 Assistant: The provided text doesn't specify the number of leaves allowed during probation.  It only states that employees are entitled to paid leave based on years of service and that all leave requires prior manager approval.
------------------------------------------------------------
```